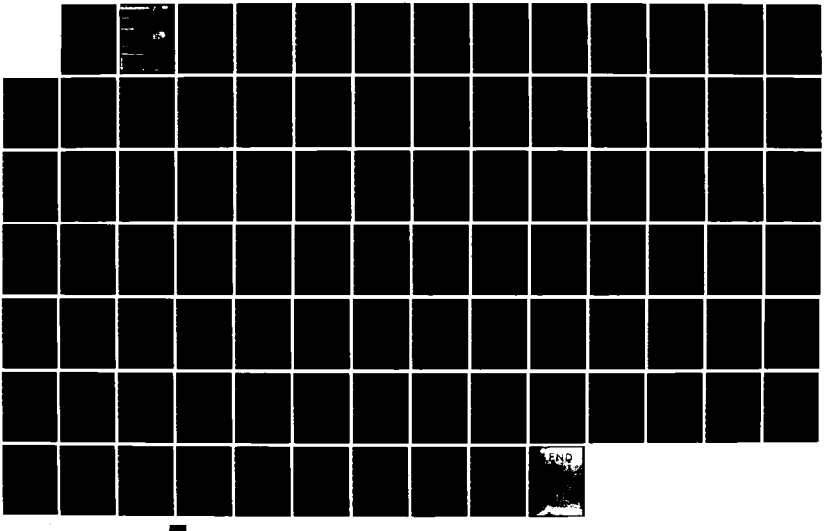
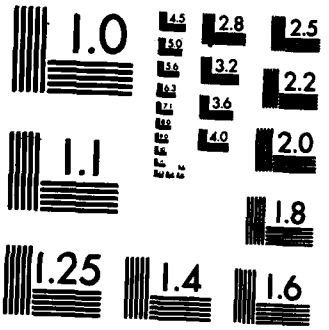
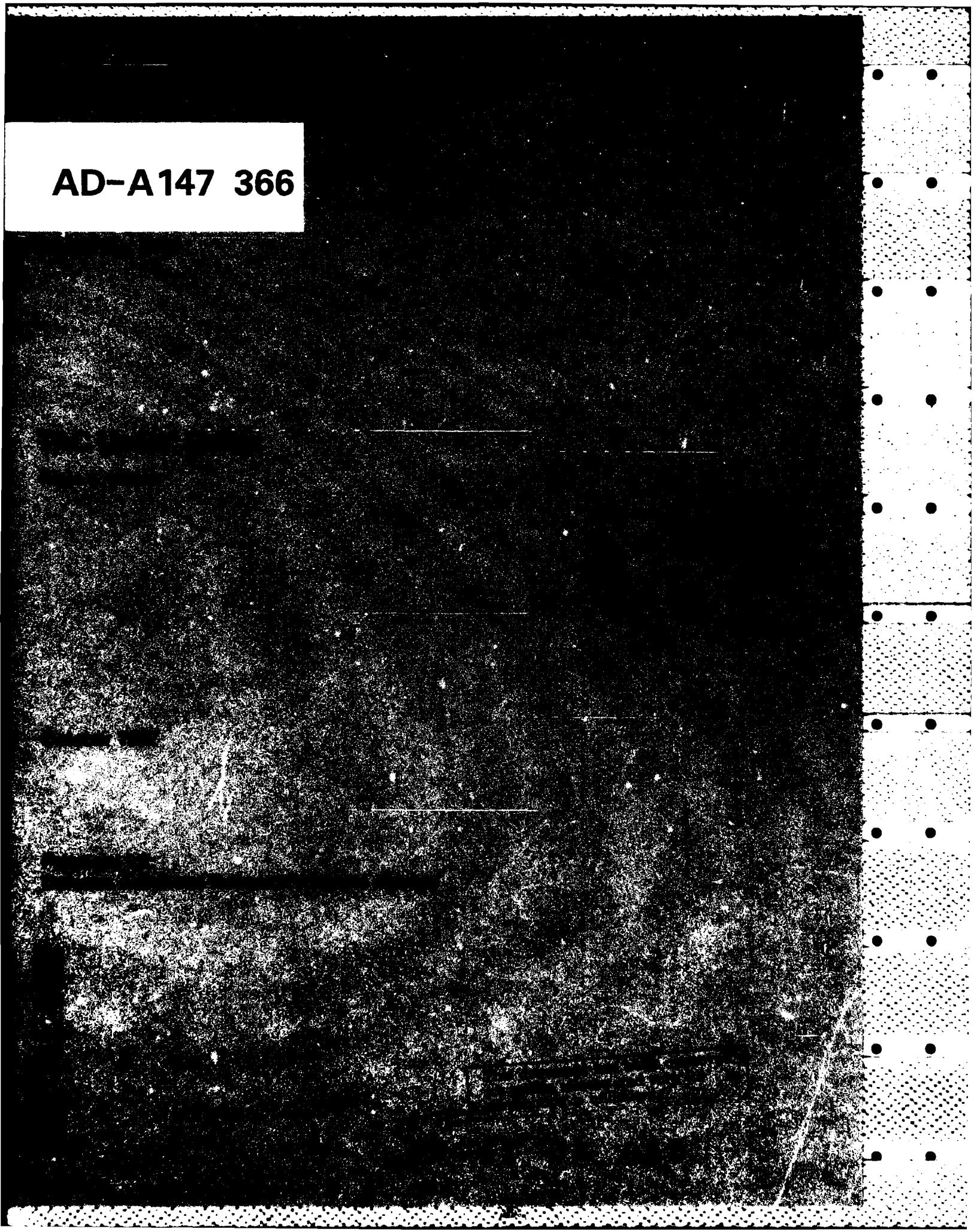
AD-A147 366 TAC (TERMINAL ACCESS CONTROLLER) USERS' GUIDE(U) BOLT BERANEK AND NEWMAN INC CAMBRIDGE MA R S CLIFFORD SEP 82 BBN-4780 DCA200-83-C-0028 1/1

UNCLASSIFIED F/G 17/2 NL




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
10.0



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* TAC Users' Guide	5. TYPE OF REPORT & PERIOD COVERED
	6. PERFORMING ORG. REPORT NUMBER BBN-RPT.4780
7. AUTHOR(s) Robin S. Clifford	8. CONTRACT OR GRANT NUMBER(s) DCA-200-83-C-0028
9. PERFORMING ORGANIZATION NAME AND ADDRESS Bolt Beranek and Newman Inc. 10 Moulton St. Cambridge, MA 02138	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Communications Agency DDN-PMO Code B626 Washington, DC 20305 (Project Monitor:John Walker)	12. REPORT DATE September 1982 13. NUMBER OF PAGES 84
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*	15. SECURITY CLASS *(of this report)* U 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

Originally written September 1981 by Robert Hinden.
Completely rewritten September 1982 by Robin S. Clifford.

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Terminal access controllers (TACS); Packet-switching; Communication networks; Users guide; Protocols; Transmission Control Protocol/ Internet Protocol (TCP/IP); Network Control Protocol.

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

The TAC Users Guide describes how to use the TAC, details the available commands and features, and discusses general information regarding the TAC. The Terminal Access Controller (TAC) system provides the hardware and software necessary to allow a user at a terminal to connect to hosts on packet-switching networks.

Originator-supplied Keywords include:

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

Unclassified

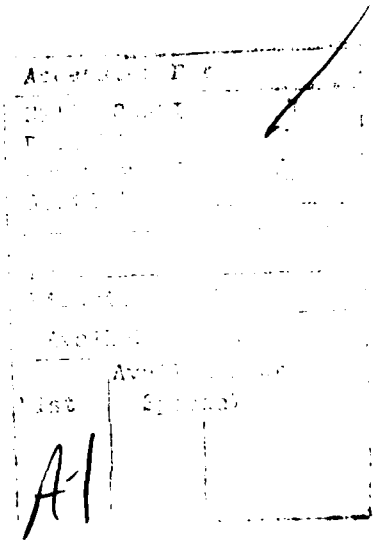
SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

BBN Report No. 4780

TAC Users' Guide

September 1982

Bolt Beranek and Newman Inc.
10 Moulton St.
Cambridge, MA 02138



UPDATE HISTORY

Originally written September 1981 by Robert Hinden.
Completely rewritten September 1982 by Robin S. Clifford.

Table of Contents

1. Introduction 1
2. Using the TAC 3
2.1 Connecting to the TAC 4
2.1.1 Hunting 6
2.1.2 Split Rate Hunting (H-316 only) 8
2.1.3 Device Rate Command 10
2.2 Giving a TAC Command 11
2.3 Sending and Receiving Data: TAC Parameters 13
3. TCP/IP and NCP Protocols 16
3.1 TCP and IP Protocols 16
3.2 TCP Operation 16
3.3 Opening a TCP Connection 17
3.4 Closing a TCP Connection 19
3.5 TCP Abnormal Conditions 20
3.5.1 Slow Reponse 20
3.5.2 TAC Rings Bells When Input is Typed 21
3.5.3 TAC Error Messages 21
3.6 NCP Protocol 23
3.7 NCP Operation 23
3.8 Opening an NCP Connection 23
3.8.1 Closing an NCP Connection 25
3.8.2 NCP Abnormal Conditions 26
4. TAC Commands 28
4.1 TAC Command Language 28
4.2 Device Rate 29
4.3 Padding and Parity 31
4.4 Linefeed Insertion 33
4.5 Transmission Characteristics 34
4.6 Clear Input Buffer 35
4.7 Intercept Character Changes 35
4.8 Protocol Selection 36
4.9 Connection Control: Open and Close 37
4.10 Reset Command 41
4.11 Connection Options: Binary and Echo Modes 42
4.11.1 Binary Input and Output Modes 43
4.11.2 Echo modes 45
4.12 Telnet Standard Control Functions 46
4.13 Flow Control Options 48
4.14 Controlling Another Port 51

4.15 Commands Requiring Authorization 53
4.15.1 Open 53
4.15.2 Wild Mode 53
4.15.3 Low Level NCP Protocol Commands 55
5. Special Operational Issues 59
5.1 Dealing with the Network Operations Center (NOC) .. 59
5.2 Site Tailoring and Default Values 60
5.3 Differences Between the TIP and the TAC 62
6. Appendix A - Command Summary 64
7. Appendix B - TAC Messages to the Terminal User 69
8. Appendix C - Connection of Terminals to the TAC 72
9. Appendix D - Device Rate Manipulation 80

10. Appendix E - Internet Addressing: Non-ARPANET-type Networks 83

TABLES

TAC Signal Allocation for EIA RS-232 Modem Connector ... 74
TAC Signal Allocation for EIA RS-232 Terminal Connector 76
TAC Signal Allocation for Current Loop Cable Connector . 78

1. Introduction

The Terminal Access Controller (TAC) system provides the hardware and software necessary to allow a user at a terminal to connect to hosts on packet-switching networks. The TAC gives the user a transparent connection to a remote host, as if he were directly connected to that host.

The TAC User's Guide describes how to use the TAC, details the available commands and features, and discusses general information regarding the TAC. This Guide may be used as a simple reference, or can be read through.

The TAC supports two host-to-host protocols: the Department of Defense standard Transmission Control Protocol/Internet Protocol (TCP/IP), and the ARPANET Network Control Protocol (NCP). In addition, the TAC supports the Initial Connection Protocol (ICP) for NCP, old and new Telnet protocols, and the 1822 host-to-IMP protocol. A user can connect to a remote host using either TCP/IP or NCP protocols. (It should be noted that the DoD intends to stop supporting the NCP protocol as of 1 January 1983.)

The TAC runs on either a Honeywell 316 computer or a BBN C/30 system. The H-316 has 32K of memory and a 63-port Multi-Line Controller (MLC). The C/30 has 64K of memory and one or two 32-port asynchronous multiplexors. (NOTE: port 0 cannot be used. The C/30 allows up to 31 or 63 ports, depending on whether it has one or two multiplexors.) The TAC is a host connected to an IMP but separate from it (unlike the older TIP which runs a logically separate program within the IMP hardware). Connection of a TAC to a host is via an 1822 hardware interface. The TAC can be either a local or distant host, but cannot, as yet, run as a VDH or HDH host. There can be more than one TAC on an IMP.

For the user, the TAC is very similar to the TIP in its operation. It supports most of the TIP's commands and features, as well as providing additional features. For information regarding the differences between the TIP and the TAC, see Section 5.3.

2. Using the TAC

The TAC is designed to allow the terminal user to communicate with a remote host as if directly connected to that host. Once the connection is established, the user will normally ignore the TAC and use the remote host.

Prior to actually connecting to the TAC, the terminal user will be concerned with terminal hardware, power requirements, modem equipment (for dial-in access), etc. Although these concerns are not directly related to the TAC, the user should be sure that the equipment chosen is compatible with the TAC. (See Appendix C - "Connection of Terminals to the TAC" - for the equipment requirements of the TAC. See Appendix D - "Device Rate Manipulation" - for limitations of device rates on the TAC.)

Once the correct equipment has been selected, the user can then go about connecting to the TAC, connecting to the remote host, and, when done, closing the connection.

The TAC maintains a set of default parameters. It is advised that the user accept these default parameters as they have proven to meet the needs of most TAC users. There are, however, a number of optional parameters available in the TAC. The user,

if need be, can instruct the TAC to execute these options to meet specific needs. (See Section 4 - "TAC Commands" - for a complete description of all commands and options available to the user.)

The rest of this section describes different uses of the TAC, how to connect to the TAC and the remote host, hunting to the proper terminal speed, etc. It is intended for general use. Section 4 - "TAC Commands" - contains detailed descriptions of specific uses of the TAC. If the user has problems with the TAC, contact the individual site liaison or the NOC (See Sectio 5.1 - "Dealing with the NOC") for assistance.

2.1 Connecting to the TAC

Once the equipment requirements have been met, the user is ready to connect to the TAC. To connect, the terminal has to be wired directly to the TAC or connected via a dial-up modem. The connection to the TAC is called a "port" and will be referred to as such throughout this guide.

When the connection is made, the user types:

<Break>

<Control-Q> (Hold the <Ctrl> key down and press the "Q" key at the same time. NOTE: the C/30 TAC cannot hunt to 9600 baud using <Control-Q>. Another character can be used instead, such as <Carriage-return>.)

The TAC prints the "herald" message:

<sitename> TAC <version #>:<port #>

or, the TAC will print a garbled message or nothing.

If the TAC prints the herald message, the port is set to the correct rate for the terminal. The user can proceed to open a connection to a remote host. If a different rate is desired, follow the procedure described later in this section.

If the TAC prints a garbled message or nothing, the terminal and the port are set at different speeds. As most TAC ports are set to hunting ("hunting" means the TAC port will recognize the terminal's speed from the first character typed and set itself to that rate), the user should try to hunt to the correct rate. This procedure is described below. If hunting is unsuccessful, contact the individual site liaison or the NOC for assistance.

Some TAC ports are preset to a known rate. The individual site liaison is responsible for such assignments and should be contacted if there are questions (see Section 5.2 - "Site Tailoring and Default Values"). If the port is set to a specific rate, change the terminal's speed to the correct rate. If there are still problems, contact the NOC.

The user may want to change the port speed, even for a preset rate port. This can be done by using the Device Rate command (see Section 4.2 - "Device Rate"). It should be noted that ports with preset rates may be protected from change. This ensures that the port will be predictably set for all users who access it.

2.1.1 Hunting

If the port the terminal is connected to is set for hunting, the user should set the terminal to the desired input and output rates. Both input and output rates must initially be the same. (For split rates and output greater than 2400 baud, see Section 2.1.2 - "Split Rate Hunting".)

Once the desired rate is set, type:

<Break>
<Control-Q>

The TAC prints:

<sitename> TAC <version #>:<port #>

Typing <Break> puts the port (if it is a hunting port) into hunting mode. If there are problems, <Break> returns the port to this hunting mode (except when there is an open connection to a remote host. If there is such a connection, <Break> will cause the TAC to send a Telnet "break" to the remote host). <Control-Q> (ASCII DC1, octal 021) is the character the TAC uses to deduce the terminal rate.

The TAC can hunt to any standard rate from 110 to 9600 baud. The <Control-Q> will set the input and output rates to the output rate deduced from the terminal. If this is sufficient, the TAC will type the herald message. The user can then continue to open a connection to a remote host, or set up any special requirements for the TAC.

For C/30 TACs, terminals using ports on the TAC must be set to the same input/output rate. Legal rates are: 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800 and 9600 baud. A terminal may hunt to any of these rates. See Section 2.1 - "Connecting to the TAC" - for information on hunting to 9600 baud.

For H-316 TACs, there is a hardware limitation that prohibits input rates to be greater than 2400 baud. The H-316 TAC, therefore, has the capability to have different input and output rates. If the user wants to have the output rate greater than 2400 baud, there are two methods of achieving this. (Both methods require switching the terminal's rates. Consult the manufacturer's documentation for the procedure.) The first method is called "split rate" hunting (see Section 2.1.2). The second method is via the Device Rate command (see Section 2.1.3).

2.1.2 Split Rate Hunting (H-316 only)

To set a port to a split rate, with output greater than 2400 baud, set the terminal speeds for input and output to the desired output rate. Then, type:

<Break>

<Control-Q>

The TAC will print:

Set Input Rate, Then Type ^Q

Set the input rate on the terminal to a rate of 2400 baud or less, leaving the output rate at its initial setting. Then, type:

<Control-Q>

The TAC will print:

<sitename> TAC <version #>:<port #>

The TAC is ready for the user to connect to a remote host, or to set up specific options in the TAC. (Note: the above method, in effect, hunts twice. First, for the output rate, then, for the input rate. <Break> is only typed at the very beginning of the process. If it is typed a second time, the port will return to its initial hunting state.)

2.1.3 Device Rate Command

To change the device rate of a port via the Device Rate command, set the terminal rate to some input/output value (2400 baud or less for input for H-316 TACs). Then, type:

<Break>

<Control-Q>

The TAC prints its herald message:

<sitename> TAC <version #>:<port #>

The user can then use the Device Rate command to change the input/output rate of the port. The command syntax is:

@DEVICE RATE # (or, @D R #)

is the parameter that specifies the rate. See Section 4.2 - "Device Rate" - in the Command section for interpretation of #.

Once the device rate has been changed for the port, set the terminal rate to the correct setting.

The Device Rate command allows the TAC port to be set to the following rates: for input, 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400 baud (plus 4800 and 9600 baud for the C30 TAC); for

output, all the above rates apply (including 4800 and 9600 baud).

2.2 Giving a TAC Command

When the user wishes to communicate with the TAC (e.g., to open a connection to a remote host, set TAC parameters, reset a port, etc.), TAC commands must be given. These commands need to be flagged in some way that the TAC can recognize. This flag is called the intercept character. The default for the intercept character is the "@" symbol. All commands to the TAC must be preceded by the intercept character for the TAC to accept them.

Once the command is typed to the TAC, it is executed with a <Carriage-return> if the user is satisfied with the command. If the user is not satisfied with the command, <rubout> will abort it.

If the user wants to include the intercept character (as text) in the data stream to the remote host, typing the intercept character twice will cause the second character to be sent to the host. For example, if the user types:

@@

the TAC will send the second "@" as data to the host. If the port is in remote echo mode (see Section 4.11.2 - "Echo Modes"), the user will see "@@@" from the TAC. The first two "@" are echoed by the TAC, the last "@" is echoed from the remote host.

Generally, commands to the TAC can be given at any time. They need not start on a new line. The TAC does not distinguish between upper or lower case letters.

Between the intercept character and the <Carriage-return>, there should be one or more words for the command, followed, perhaps, by one or more parameters. The TAC only recognizes the first letter of each word in a command. Therefore, commands can be abbreviated. For example, if opening a connection to a remote host, the full command would be:

@OPEN 10:2/49 <Carriage-return> (TCP only)

@OPEN 2/49 <Carriage-return> (TCP or NCP)

The TAC will also recognize:

@O 10:2/49 <Carriage-return> (TCP only)

@O 2/49 <Carriage-return> (TCP or NCP)

Commands can consist of several words. For example:

@DEVICE CODE ASCII <Carriage-return>

This command can be abbreviated to:

@D C A <Carriage-return>

The spaces are required. @DCA <Carriage-return> would be interpreted by the TAC as @D, and is not valid.

2.3 Sending and Receiving Data: TAC Parameters

Once a connection to a remote host is open, sending data to that host is straightforward. The TAC permits the user to communicate with the host as if directly connected. The TAC saves 7-bit ASCII characters in an input buffer and transmits them at appropriate intervals in messages to the host. Several commands exist in the TAC to control the frequency of transmission of characters (see Section 4.5 - "Transmission Characteristics").

The TAC does not perform code translation. The single exception (in accordance with Telnet protocol) is that a <Carriage-return> is automatically translated to <Carriage-

return><Linefeed> or <Carriage-return><Null>. The default is <Carriage-return><Linefeed>. Some TAC ports may be preset to <Carriage-return><Null>. For further discussion, see Section 4.4 - "Linefeed Insertion".

As mentioned earlier, the TAC interprets the "@" symbol to be the command intercept character. If the "@" is a frequently used character, the user may want to change the intercept character to something else. This is described in Section 4.7 - "Intercept Character Changes".

Telnet protocol defines several standard representations for control functions, such as "break" and "interrupt process". These are intended as a convenience to the user of several different host systems. The user is thus not required to learn the conventions of each host. The TAC provides several commands to insert these standard function codes in the data stream. These commands are described in Section 4.12 - "Telnet Standard Control Functions".

Receiving data from the remote host is also straightforward. The TAC receives characters and sends them to the terminal without translation.

If the user desires complete transparency on input and/or output (all eight bits, no intercept character, parity or fill characters), the TAC allows 8-bit binary mode - if the host agrees. Section 4.11 - "Connection Options" - describes binary mode more fully.

As stated earlier, the user is advised to accept the default parameters of the TAC. The novice user will be able to use the TAC with a minimum of experience. However, if the experienced user has more complex needs, the TAC has the versatility to satisfy them.

3. TCP/IP and NCP Protocols

3.1 TCP and IP Protocols

The Transmission Control Protocol (TCP) and the Internet Protocol (IP) are highly reliable protocols for host-to-host communication within a packet-switched computer network and between such networks.

The TCP/IP protocols have been declared, by the Department of Defense, as the basis for a DoD standard host-to-host communications protocol. Support for the ARPANET NCP (Network Control Protocol) will be withdrawn sometime in the future.

3.2 TCP Operation

A TCP connection consists of opening a connection to a remote host, sending and receiving data, and closing the connection. However, it should be kept in mind that TCP will attempt to keep a connection to a host active even if that host does not acknowledge data from a TAC or is down. TCP will try to send data to a host about once a minute or until the

terminal user tells the TAC to stop connection attempts.

3.3 Opening a TCP Connection

There are two ways to open a connection to a host via TCP. The first uses a "net#:host#/IMP#" combination. The second method utilizes a four-field internet address and is only permitted in TCP. It is described in Section 4.9 - "Connection Control".

The first method is used as follows:

User types: @OPEN net#:host#/IMP# <Carriage-return>

TAC prints: "TCP Trying..."

The TAC will print one of the following:

Open
: The remote host agrees to connect. The user may then continue.

Destination host dead
: Remote host is not communicating with the network.

Host down until <day> at <hour>:<minutes> <timezone>.
: Remote host is not communicating with the network. The day and time when the host was most recently scheduled to come up is indicated.

Refused
: The remote host rejected the attempt to establish

a connection. This may occur if the host does not support TCP and/or Telnet. The TAC port will now be idle.

Retransmitting
This indicates that TCP is having to retransmit many times in attempting to open a connection to a host. The host may not support TCP and is ignoring the connection attempt.

Destination Unreachable
There is no path from the TAC to the remote host through the communication networks.

Can't
The user already has an active connection to another host. It is necessary to close or reset the connection before another attempt can be made.

Open error
An error occurred while the OPEN attempt was in progress. This probably indicates a host error and should not happen often. If it is recurrent, contact the NOC (see section 5.1)

TAC's IMP down
The IMP that connects the TAC to the network is down.

A "connection" consists of a data path in both directions, between the TAC and a remote host. When the "OPEN" occurs, the connection is complete. A port may only have one open connection at a time.

3.4 Closing a TCP Connection

Before a host connection is closed, the user must first log out of the remote host. Follow whatever procedures the host requires. Then, to close the connection between the TAC and the host, type:

@CLOSE <Carriage-return>

The TAC will print "Closed" when the connection to the host is terminated. The host has agreed to end the connection in both directions. The "Closed" message should appear in a second or two. However, sometimes the host does not respond to the @CLOSE command. If this happens, the user should reset the connection by typing:

@RESET <Carriage-return>

Once a connection is closed or reset, the user can then open a new connection.

3.5 TCP Abnormal Conditions

Once a connection is open, the TAC (and network) will normally provide an invisible link to a remote host. However, problems will sometimes occur, such as slow response, pauses, or broken connections. Some of the most common problems are described below.

3.5.1 Slow Reponse

Slow response occurs for a number of reasons but is usually transient. It can be caused by heavy load on some part of the network, the host, or the TAC. Or, a line or node down on the network can cause heavy load on alternate routes, thus causing slow response. If the problem does not clear within a short period of time and is not host related, contact the NOC (see Section 5.1 - "Dealing with the NOC").

3.5.2 TAC Rings Bells When Input is Typed

If the TAC's buffers are full when a character is typed, it will discard the character and ring the terminal's "bell" by sending <Bell> (ASCII BEL, octal 007). (<Bell> will also be sent when Telnet negotiations fail.) The buffers will be full only if congestion exists on the network or in the host, preventing the TAC from sending buffered characters. If the problem persists for several minutes, the user can try to flush the buffers using the @FLUSH and/or @SEND SYNC commands. (See Section 4.6 - "Clear Input Buffer" - and Section 4.12 - "Telnet Standard Control Funtions".)

3.5.3 TAC Error Messages

In the event of problems with the connection between a terminal and a remote host, the TAC attempts to keep the user informed. The TAC will give the following error messages if a problem should arise:

Retransmitting
: TCP has to retransmit many times to get the TAC data accepted by the remote host. The message will appear after TCP has retransmitted the data five times. It will then appear

about once a minute until the host accepts the data or the TAC port is reset.

Host closing connection
The remote host has closed its connection to the TAC. The TAC will close its connection to the host. The TAC port will then be idle.

Destination host dead
The remote host is not communicating with the network.

Destination unreachable
No path exists from the TAC to the remote host through the communications networks.

Host down until <day> at <hour>:<minutes> <timezone>.
The remote host is not communicating with the network. The day and time that the host was most recently scheduled up is indicated.

Host reset connection
The remote host reset the connection to the TAC. The TAC port is now idle.

TAC going down in <mins> mins for <hrs> hrs <mins> mins.
The TAC is going down in the time indicated.

TAC going down NOW
The TAC will go down immediately.

TAC's IMP going down in <mins> mins for <hrs> hrs <mins> mins
The TAC's IMP is going down in the time indicated. Although the TAC will still respond, it will be isolated from the network.

TAC's IMP going down NOW
The TAC's IMP will go down immediately.

TAC's IMP down
The TAC's IMP is down. The TAC is isolated from the network.

<sitename> TAC <version #>:<port #>

The TAC has restarted. The connection has been reset.

3.6 NCP Protocol

The Network Control Protocol (NCP) is a host-to-host communication protocol used on the ARPANET. It is limited to use on the ARPANET and will be replaced by TCP in the near future (scheduled date - 1 January 1983).

3.7 NCP Operation

NCP operation is very similar to TCP. However, it is not as reliable or versatile as TCP and will close a connection to a remote host when problems are encountered.

3.8 Opening an NCP Connection

Opening a connection to a remote host using NCP is the same as with the first method for TCP (see Section 3.3 - "Opening a TCP Connection"). It cannot be used except on ARPANET hosts supporting NCP.

To open a NCP connection, the user types:

@OPEN host#/IMP#

The TAC prints "NCP trying...". The user will get one of the following responses:

Open
The remote host agrees to connect. The user may continue.

Destination host dead
The remote host is not communicating with the network.

Host down until <day> at <hour>:<minutes> <timezone>.
The remote host is not communicating with the network. The day and time when the host was most recently scheduled to come up is indicated.

Refused
The remote host has rejected the connection attempt.

Destination unreachable
There is no path from the TAC to the remote host through the communications network.

Can't
The TAC already has a connection to a remote host. The user must close or reset the connection before another attempt can be made.

Open error
An error occurred while the OPEN attempt was in progress. This probably indicates a host error and should not happen often. If it is recurrent, contact the NOC (see section 5.1).

TAC's IMP down
The IMP that connects the TAC to network is down.

In any case except the OPEN, the TAC port will be idle after receiving the above responses.

3.8.1 Closing an NCP Connection

To close an NCP connection, type:

@CLOSE <Carriage-return>

The TAC will print "Closed". The TAC port is now idle.

To reset an NCP connection, type:

@RESET <Carriage-return>

The TAC will print:

<sitename> TAC <version #>:<port #>

Once a connection is closed or reset, the user can open a new connection.

3.8.2 NCP Abnormal Conditions

Problems with an NCP connection to a remote host are similar to those using TCP. Error messages for NCP are listed below. For other types of abnormal conditions, see Section 3.5 - "TCP Abnormal Conditions".

Closed
The remote host has closed the connection.

Destination host dead
The remote host is not up to the network.

Host reset connection
The remote host has reset the connection to the TAC.

Host down until <day> at <hour>:<minutes> <timezone>.
The remote host is not communicating with the network. The day and time that the host is scheduled to come up is indicated.

TAC going down in <mins> mins for <hrs> hrs <mins> mins
The TAC is going down in the time indicated.

TAC going down NOW
The TAC will go down immediately.

TAC's IMP going down in <mins> for <hrs>hrs <mins> mins
The TAC's IMP is going down in the time indicated. Although the TAC will still respond, it will be isolated from the network.

TAC's IMP going down NOW
The TAC's IMP will go down immediately.

TAC's IMP down
The TAC's IMP is down. The TAC is isolated from the network.

<sitename> TAC <version #>:<port #>
The TAC has restarted. The connection has been reset.

4. TAC Commands

Some of the TAC commands available to the user were covered in Section 2 - "Using the TAC". These and other commands will now be explained in more detail. This section is for reference and should be used to clarify the options available on the TAC. The commands are local to the TAC and should not be confused with host-level commands.

4.1 TAC Command Language

In order to communicate with a TAC, it is necessary to flag a command with a character the TAC can identify. This is called the intercept character. The "@" symbol is the default intercept character, but can be changed to another character by using the "intercept" command (see Section 4.7 - "Intercept Character Changes").

The command itself is made up of words separated by spaces. Commands can be abbreviated, since the first character of each word is the only one recognized by the TAC. Upper or lower case letters can be used.

Some commands include one or more parameters following the command text. These parameters are in decimal, with the exception of the NCP socket number, which is in octal.

Once typed, the command is executed with a <Carriage-return>. A command can be aborted by using <Rubout> (ASCII DEL, octal 177) or by typing an illegal command. The TAC will print "Bad" or "Can't". Most commands can be given at any time and can begin anywhere on a line.

If the intercept command is doubled (i.e., typing "@@" or two of the selected intercept characters) it is treated as a single text character. It will be transmitted to the host as normal text.

4.2 Device Rate

Although the TAC can hunt to a variety of baud rates, it is sometimes useful to set a port to a specific speed. For example, many lineprinters are receive-only devices and are unable to transmit their baud rate to the TAC to set the port speed.

The TAC supports the following rates: input and output - 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800 and 9600. (Note: the Honeywell 316 TAC can only run up to 2400 baud for input, but can run split speeds. The C/30 TAC must have input and output the same.) The TAC also supports externally clocked rates (terminal provides the clock). Rates may be pre-set in an individual site's parameter file (see Section 5.2 - "Site Tailoring and Default Values"). A port may have its rate specified as permanent, in which case it cannot be changed via user command.

To set the device rate for a port, type:

@DEVICE RATE # (@D R #)
is a decimal number representing a 10-bit field. It sets the input rate, output rate and character size. Since most ASCII terminals use 8-bit characters, the following list gives examples of available rates with 8-bit characters.

Decimal Parameter	I/O Rate
113	75 baud
178	110
243	134.5
308	150
373	300
438	600
503	1200
568	1800
633	2400
698	4800 (C/30 TAC only)
763	9600 (C/30 TAC only)
754	110 input/9600 output (H-316 TAC only)
757	300 input/9600 output (H-316 TAC only)
761	2400 input/9600 output (H-316 TAC only)
1023	Externally clocked

(Note: See Appendix D for more information regarding rates and character size.)

4.3 Padding and Parity

Some terminals and lineprinters need extra time after <Carriage-return> <Linefeed> for the print head to return to margin. Others, CRTs for example, need no padding. The TAC provides several options for padding output to a terminal.

Certain terminals, especially the Teletype Model 37, require even parity. The TAC supports even parity or no parity. The default on the TAC is no parity.

The default for the TAC is to insert no padding and no parity on output. (The exception is: when the TAC hunts to 300 baud, it automatically sets extra padding.) The options available are to insert padding on output to slow carriage-return terminals or lineprinters and to set even parity. The desired option can be pre-set in the site parameter file (see Section 5.2 - "Site Tailoring and Default Values"). It should be noted that binary output mode suppresses any padding and parity.

To select the desired padding or parity, type:

@DEVICE CODE ASCII (@D C A)
The TAC will not insert padding on output to the terminal. Any prior padding is disabled.

@DEVICE CODE EXTRA-PADDING (@D C E)
The TAC pads the output with <Nulls> after a <Carriage-return> is sent by the terminal. The number of <Nulls> sent is related to the terminal's output rate. The faster the rate, the more <Nulls> inserted. This option is useful with terminals that have a slow carriage-return, such as Texas Instrument or Execuport terminals.

@DEVICE CODE OTHER-PADDING (@D C O)
The TAC pads output to an ODEC lineprinter. (This brand printer requires special timing for

output.)

@DEVICE CODE 37 (@D C 37)
The TAC sets the parity of the port to even parity.

To change any of the padding or parity settings, the user must send an @D C A first. Also, padding and parity cannot be set when flow control is enabled.

4.4 Linefeed Insertion

When a terminal sends a <Carriage-return>, Telnet protocol requires the TAC to send a <Linefeed> or a <Null> character. The default for the TAC is to insert <Linefeed>. Linefeed insertion is suppressed in binary input mode. To set the character insertion, type:

@INSERT LINEFEED (@I L)
TAC inserts <Linefeed> after the terminal sends <Carriage-return>.

@CLEAR INSERT LINEFEED (@C I L)
TAC inserts <Null> after the terminal sends <Carriage-return>.

4.5 Transmission Characteristics

Once a connection to a remote host is established, the TAC normally sends every character to the host as soon as it is typed. However, the user can specify when the TAC should send data. The following commands represent the options available. They are maximum values, since the TAC may send data sooner than specified if its input buffer fills up.

To change the transmission rate, type:

@TRANSMIT EVERY # (@T E #)
TAC attempts to send every # characters. # is a decimal number. The initial TAC mode is TRANSMIT EVERY 1.

@TRANSMIT ON LINEFEED (@T O L)
TAC transmits to the remote host whenever a <Linefeed> is received from the terminal.

@TRANSMIT ON MESSAGE-END (@T O M)
TAC transmits to the remote host whenever a <Control-S> is received from the terminal.

@TRANSMIT NOW (@T N)
TAC immediately sends all characters stored in input buffers to the remote host. This does not change other transmission modes previously set. The TAC may always be forced to transmit the contents of its buffers by using the @T N command.

@TRANSMIT EVERY 0 (@T E 0)
Used to reset the TRANSMIT ON MESSAGE-END and the TRANSMIT ON LINEFEED modes.

These modes will not transmit until the right character is received, or the buffer is full, or an @T N is sent.

4.6 Clear Input Buffer

@FLUSH (@F)
TAC clears any unsent characters in the input buffers.

4.7 Intercept Character Changes

Although the "@" is the default intercept character, it is sometimes useful to change it to an alternate character. (NOTE: the intercept character may be specified as permanent in the site file and thus be unchangeable via user command. See Section 5.2 - "Site Tailoring and Default Values".

To change the intercept character, type:

@INTERCEPT # (@I #)
is the decimal number representing the ASCII character desired.

Some of the common intercept characters are listed below along with their decimal value:

Decimal value	ASCII character
004	EOT <Control-D>
016	DLE <Control-P>
025	EM <Control-Y>
033	!
035	#
037	%
039	'
064	@
092	\
094	^
095	_ (Underscore)
124	|
126	~ (Tilde)

@INTERCEPT ESC (@I E)
Sets intercept character to "@". "@" is the default.

@INTERCEPT NONE (@I N)
No intercept character is recognized. The port is in a pseudo "7-bit binary mode". Once this mode is enabled, a reset must be given from another port on a TAC to disable this mode on the TAC port.

4.8 Protocol Selection

The protocols available to the user are TCP or NCP. Unless specified in the site parameter file, the default is TCP.

To select the desired protocol, type:

@PROTOCOL TCP (@P T)

The TAC sets the port protocol to TCP.

@PROTOCOL NCP (@P N)
The TAC sets the port protocol to NCP.

The protocol may be changed only when no open connection to a remote host exists.

4.9 Connection Control: Open and Close

The Open and Close commands permit the user to establish or terminate a connection to a remote host. A list of the messages the TAC will print can be found in Appendix B - "TAC Messages to the Terminal User".

There are two types of Open commands. The first type can be used in TCP or NCP, as follows:

@OPEN net#:host#/imp#;port# (@O net#:host#/imp#;port#)*

The TAC will open a connection to the specified remote host.

The "net#:" field is the number of the network to which the remote host is connected. It can only be used in TCP mode. The number can be from 0 to 255 and is decimal. The default is the

* ";port#" is a TCP-specific term and should not be confused with the term "port" which refers to the TAC port to which the user's device is attached.

network the TAC is on. "net#:" is not given when in NCP mode.

The "host#/" field is the number of the remote host on the IMP. The number can be from 0 to 255 and is decimal. If not specified, the default is zero. In TCP mode, the "host#/" field is mapped in the upper 8-bits of the "local address" portion of the internet address.

The "imp#" field is the IMP number on which the remote host is connected. There is no default and it must be specified. The number can be from 0 to 65,535. It is decimal. In TCP mode, the "imp#" field is mapped in the lower 16-bits of the internet "local address".

The ";port#" field is the number of the port or socket the TAC will connect to. This number is decimal in TCP mode and octal in NCP mode. In either mode it is optional. The default value is the new Telnet socket (23 decimal, 27 octal). In TCP mode, it is the 16-bit port that the TAC will connect to. In NCP mode, it is the 32-bit socket that ICP will use as a logger socket.

The following are all valid examples of connecting to the Telnet socket (23) on host 0 of IMP 5 on network 10:

```
@O 10:0/5;23 (TCP only)

@O 0/5;23

@O 5;23

@O /5

@O 10:/5
```

The other type of Open command available can only be used in TCP mode. It is formed as follows:

@OPEN a.b.c.d;port# (@O a.b.c.d;port#)

For ARPANET-type networks, the formula is defined as:

"a." is the network number. For example, the ARPANET number is 10. The default is the network the TAC is on.

"b." is the host number on the IMP. The default is zero.

"c." is the logical host number. The default is zero.

"d" is the IMP number the remote host is connected to. There is no default. The IMP number must be given.

";port#" is the TCP port to which the connection will be made. It is optional. It defaults to Telnet (23). It is a decimal number.

(NOTE: if the default is desired as a parameter, it is sufficient to just enter the period after a parameter. For example, @O 10...5 is a legal address.) **

** Other valid examples of the Open commands using TCP are:

```
@O ...5
@O ..5
@O .5
@O 5
```

All open a connection to host 0 on IMP 5.

For non-ARPANET type networks, the parameters "a.b.c.d" specify a 32-bit Internet address. See Appendix E for a complete breakdown of the Internet address.

If a user wishes to communicate between TAC ports, the TCP ";port#" number or the NCP socket number has to be calculated for the remote port.

To define the ";port#" for TCP, the formula is:

(decimal)port # X 256 + 23 = TCP ;port#

The NCP socket number is calculated by putting double the octal port number in the left 16-bits and a 2 or 3 (transmit or receive) in the right 16-bits. See Section 4.14.4 - "Low Level NCP Protocol Commands" - for an example of the NCP socket number.

To terminate a connection, use the Close command. It applies to both TCP and NCP connections.

To close a connection to a remote host, type:

@CLOSE (@C)
The TAC connection to the remote host is terminated. The TAC port is now idle.

4.10 Reset Command

The Reset command will restore the TAC port to its initial state. It applies to both TCP and NCP modes. It is especially useful, however, when the port is in TCP mode. If a remote host goes down or becomes unreachable, TCP will keep retransmitting to that host about once a minute. The CLOSE command will not be acknowledged. In this event, the Reset will close the connection completely. The port will then be ready for any other connection attempts.

To reset a TAC port, type:

@RESET (@R)
The TAC closes any existing connections. Terminal parameters are returned to their preset values. The TAC will print the herald. This command does return the port to a hunting state.

The Reset command will clear certain options that have been previously set on a port. Reset will do the following:

- Simultaneously do a "give back" on the user's port, the port the user is controlling and will force any other port to relinquish control. (See Section 4.14.3 - "Controlling Another Port".)
- Returns echo mode to LOCAL ECHO, with REMOTE ECHO desired (when connected to a remote host).
- Sets port to allow 8-bit binary mode when requested.

- Sets transmission mode to TRANSMIT EVERY 0.
- Returns "wild" mode to not wild.
- Inserts <Linefeed> after <Carriage-return>.

Reset does not change the Device Rate setting for a port. That rate will remain the same as it was before the Reset command was given. Reset does not alter flow control, padding or parity.

4.11 Connection Options: Binary and Echo Modes

Telnet protocol allows the user (TAC) and server hosts to negotiate transmission characteristics and define a default set of parameters. It then permits each side of the connection to negotiate for non-standard options. Hosts are not required to accept non-standard options and can refuse to accept them without understanding what they have just refused. This permits sophisticated hosts to define complex options without burdening simple hosts.

The TAC provides two options for the user. These options are Binary (input and/or output) Mode and Remote Echo Mode. The TAC to host connection must exist to use these options.

Note, also, that a request for these options does not guarantee acceptance.

4.11.1 Binary Input and Output Modes

Normally, the TAC sends and receives data to and from a remote host in 7-bit characters. The Binary Mode option allows all eight bits to be sent or received over a connection to the remote host. Usually, the TAC user will not need the Binary Mode option.

When binary mode is requested, the TAC sets an internal flag for the mode the user desires. It will then attempt to get the remote host to agree to implement the mode. It is important to note that a connection to a remote host must exist for the binary options to function. The TAC will passively accept a binary mode request from a remote host. (NOTE: If the user sets @B I E or @B O E (see below) on a TAC port, the port will reject any requests for binary mode from a remote host or a remote terminal on a TAC.)

The commands for binary mode(s) are:

@BINARY INPUT START (@B I S)
Enables eight-bit binary input mode.

@BINARY INPUT END (@B I E)
Disables eight-bit binary input mode.

@BINARY OUTPUT START (@B O S)
Enables eight-bit binary output mode.

@BINARY OUTPUT END (@B O E)
Disables eight-bit binary output mode.

When Binary Mode(s) are in effect, the TAC passes all eight bits of data to and/or from the remote host. In Binary Output Mode, there is no padding or parity added to the output. In Binary Input Mode, <Linefeed> and <Null> are not inserted after <Carriage-return>. NOTE: the intercept character and local TAC commands cannot be given with binary input mode enabled. The remote host or a remote terminal on a TAC must terminate binary input mode on the port.

Although binary input and output are independent in the TAC, some hosts only negotiate both together. Since binary mode can only be executed on an open connection, closing the connection returns the TAC port to non-binary mode and allows local command execution.

4.11.2 Echo modes

Echoing is the process of outputting characters typed as input by the user back to the terminal. This process can occur at the terminal (ECHO HALFDUPLEX), in the TAC (ECHO LOCAL), or in the remote host (ECHO REMOTE).

The commands to set echo mode are:

@ECHO HALFDUPLEX (@E H)
: The TAC assumes the terminal is physically providing its own echo of characters. The TAC will not echo input characters from the terminal and will attempt to negotiate local echo with the remote host. The TAC will send to the terminal any characters it inserts in the data stream (<Linefeed> after <Carriage-return>, for example). This mode is used for physical halfduplex terminals or special-purpose terminals connected to the TAC to eliminate almost all echoing.

@ECHO LOCAL (@E L)
: The TAC echoes all characters received from the terminal.

@ECHO REMOTE (@E R)
: If a connection to a remote host is open, the TAC requests the host to echo. The TAC will still echo TAC-level commands.

The default for the TAC, when not actively connected to a host, is ECHO LOCAL. When a connection is active, the default changes to ECHO REMOTE. Remote echo requires that a TAC-to-host connection exist. Most users will not need to use the echo

commands. The TAC does not support the Telnet Remote Controlled Transmission and Echoing (RCTE) option.

4.12 Telnet Standard Control Functions

The function of the Telnet protocol is to provide a standard interface for terminal devices and terminal-oriented processes throughout the network. The Telnet protocol defines a standard representation for certain functions, described below. These representations have standard, but not required, meaning. (NOTE: a system that does not provide such functions to local users need not provide them to network users. The standard representation for a function may be treated as a no-operation by the host.) A system that provides the function to the local users is obliged to provide the same function to the network user transmitting the standard representation for the function. The user may still use the server host's method of invoking the function. The TAC commands simply send the appropriate Telnet function codes to the host if the connection is open (or opening).

@SEND ABORT OUTPUT (@S A O)
(AO) - Many systems provide a function that allows

a process, which generates output, to run to completion without sending the output to the user's terminal. This function also clears any output already produced but not yet output to the user's terminal. AO is the standard representation for implementing this function. The server host should allow the current process to run, or appear to run, to completion. It will not send its output to the user terminal. A SYNC is also sent to the user.

@SEND ARE YOU THERE (@S A Y T)
(AYT) - Many systems provide a function that gives the user some visible (i.e., printable) or audible evidence that the host is up and running. The user can implement this function if the host system is unexpectedly silent for a long period of time due to heavy load, unanticipated long computations, etc. AYT is the standard representation for this function. When the function is implemented, the server host should send back to the NVT (network virtual terminal) some visible or audible evidence that the AYT was received.

@SEND BREAK (@S B)
(BRK) - The interpretation of the BREAK function is entirely up to the host. Many server hosts ignore it. The function is provided because it is a signal outside the ASCII set and is given local meaning in many host systems. It indicates that the Break or Attention key was hit.

@SEND ERASE CHARACTER (@S E C)
(EC) - The server host may provide a function to delete the last preceding character from the stream of data from the user. The function is used to edit keyboard input when typing mistakes are made. EC is the standard representation for this function.

@SEND ERASE LINE (@S E L)
(EL) - The server host may provide a function to delete all the characters from the data stream

back to, but not including, the last <Carriage-return><Linefeed> sequence sent over the Telnet connection. This function is used to edit user keyboard input. EL is the standard representation for this function.

@SEND INTERRUPT PROCESS (@S I P)
(IP) - The server host may provide a function to suspend, interrupt, abort, or terminate the operation of a user process. This function is useful if a user believes the process is in an unending loop or if an unwanted process is inadvertently activated. IP is the standard representation for this function. The TAC automatically follows IP with a SYNC character. This function is not yet supported in TCP mode.

@SEND SYNC (@S S)
The NVT has a key labelled "SYNC". Real terminals do not have such a key. The function is unique to network use. The "SYNC" key is a cue to the server host that there is an important message buffered in an inaccessible place. The TAC and the server host go to the trouble to get the SYNC indication over a different channel to bypass the normal buffering conventions. Normal use of this function would be @S B followed by @S S.

4.13 Flow Control Options

Certain devices, such as lineprinters, need a means of preventing their internal buffers from over-flowing when data is sent from the TAC at a rapid rate. Likewise, the TAC may need a means of keeping its buffers from over-flowing when data is sent at too rapid a rate from a device (such as a terminal with

internal buffer transmission capability or a micro-computer) connected to the TAC. These requirements are accomplished by the use of input and output flow control.

The TAC provides XON-XOFF flow control for the input (terminal to TAC) and output (TAC to terminal) directions. The XOFF flow control character is <Control-S> (ASCII DC3, octal 023). The XON flow control character is <Control-Q> (ASCII DC1, octal 021).

With Output Flow Control enabled, the TAC will stop sending output to a port when it receives an XOFF from the terminal. The TAC will start sending data when it receives an XON from the terminal. This is normally used for lineprinters and terminals which provide slow scrolling of output.

With Input Flow Control enabled, the TAC will accept data from the port until the TAC's input buffer is approximately 85% full. The TAC will then send an XOFF to the port to signal the port to stop sending data. After the TAC's input buffer empties to about 70%, the TAC will send an XON to the port to signal the terminal to resume sending data.

NOTE: when the TAC sends the XOFF to the port, it expects the port to stop sending input within a few characters. If the terminal continues sending, the TAC's input buffer may overflow, causing characters to be lost. To prevent lost characters during high speed transmission through the TAC, the user should have the remote host NOT echo the data back to the user. This can be accomplished via the remote host, or, if that is not possible, by setting the TAC port to ECHO HALF-DUPLEX during the high speed transmission of data. This will prevent characters from being lost due to delays caused by the remote host's echoing of characters sent.

The commands for Input Flow Control are:

@FLOW-CONTROL INPUT START (@F I S)
Enables flow control from the terminal to the TAC.

@FLOW-CONTROL INPUT END (@F I E)
Disables flow control from the terminal to the TAC.

The commands for Output Flow Control are:

@FLOW-CONTROL OUTPUT START (@F O S)
Enables flow control from the TAC to the terminal.

@FLOW-CONTROL OUTPUT END (@F O E)
Disables flow control from the TAC to the terminal.

Certain TAC options will cause a deadlock if set while a port has flow control enabled. Therefore, the following options cannot be set at the same time: Input Flow Control and Output Flow Control; Input Flow Control and Binary Output Mode; Output Flow Control and Binary Input Mode; and, Output Flow Control and Transmit on EOM (which uses <Control-S>). Padding and parity cannot be set when flow control is enabled. Also, when Input Flow Control is enabled, the TAC will not output, to the terminal, the characters <Control-S> or <Control-Q> sent from the remote host. When Output Flow Control is enabled, <Control-S> and <Control-Q> cannot be entered into the data stream from the terminal.

Flow control can be permanently enabled for a port in the site parameter file (see Section 5.2 - "Site Tailoring and Default Values"). If it is not enabled in the site file, hanging up a data set will disable any flow control mode.

4.14 Controlling Another Port

Normally, the terminal user only sends commands to the TAC port being used. However, it is sometimes necessary to give commands from one port to another. In the case of a hung port

or a receive-only port, it is useful if the user of another port can send commands to it.

To send commands to another port, the command has to be preceded by the number of that port. The port number must be in octal. For example, to send a command to port 16 (octal), type:

```
@16 DEVICE RATE 633   (@16 D R 633)
     The device rate for port 16 would be set to 2400
     baud for input and output rates.
```

Since only one port at a time can control another port, it is necessary to give up control when finished sending commands to the other port.

To give up control of another port, type:

```
@# GIVE BACK   (@# G B)
     # is the octal number of the port one wishes to
     stop controlling.
```

NOTE: To issue a command requiring authorization, the receiving port must have that authorization set.

4.15 Commands Requiring Authorization

Certain ports, due to regulatory requirements, need to have authorization given to allow certain parameter changes from another port on a TAC. This authorization has to be approved by the individual site liaison and executed by the NOC. (See Section 5 - "Special Operational Issues".)

The following commands require authorization on the port of the TAC to which such commands are directed.

4.15.1 Open

Authorization is required to set the ";port#" parameter in the OPEN command. See Section 4.9 - "Connection Control: Open and Close" - for a complete discussion of the OPEN command.

4.15.2 Wild Mode

Sometimes it is useful for a remote host to be able to access a passive device connected to a port on the TAC (such as a lineprinter or tape punch). Normally, the TAC will refuse such

an access attempt. To permit such an attempt, the port has to be put into a state called "wild".

"Wild" mode allows a remote host to send to a selected port. There are several things one must be remember when using a port set to "wild". The remote host must know the location of the wild port: its port or socket number and host address. The request to send to the port must use the correct protocol: if the port is in NCP mode, it will not accept a TCP request; if it is in TCP mode it will not accept an NCP request.

"Wild" mode can be set via user command or can be pre-set in the site parameter file (see Section 5.2 - "Site Tailoring and Default Values"). It is useful, for example, if the remote host uses the port as an output device for spooling a listing queue to a lineprinter.

The commands for enabling and disabling "wild" mode are:

@SET DEVICE WILD (@S D W)
Enables "wild" mode on a port.

@CLEAR DEVICE WILD (@C D W)
Disables "wild" mode on a port.

A TAC port in "wild" mode will accept a connection made to its port or socket. TCP uses a 16-bit port number. NCP uses

a 32-bit socket number. For TCP, the TAC assigns TCP port numbers using the upper 8-bits set to the device port number (1-63) and the lower 8-bits set to 23. For NCP, the TAC assigns a socket number with double the octal port number in the high-order 16-bits and 2 (or 3) set in the low-order 16-bits (for the sending or receiving socket). (See the end of Section 4.9 - "Connection Control: Open and Close" - to formulate TCP ;port# and NCP socket #.)

4.15.3 Low Level NCP Protocol Commands

There are several low-level NCP commands that allow manual NCP host-to-host protocol. These commands can be used to connect to a socket other than the logger socket, or to connect to another TAC port. The commands are valid only in NCP mode, and can only be used when no connection is active. Most users will not use these commands. (NOTE: when DoD stops support of NCP, these commands will be invalid.)

The commands available are:

@HOST # (@H #)
is the network address of the host or port.

This command simultaneously initiates "@S T H" and "@R F H"

@INITIAL CONNECTION PROTOCOL (@I C P)
Starts the NCP initial connection protocol.

@PROTOCOL BOTH (@P B)
Simultaneously sets NCP "@P T T" and "@P T R".

@PROTOCOL TO RECEIVE (@P T R)
Initiates receive side of NCP connection protocol.

@PROTOCOL TO TRANSMIT (@P T T)
Initiates transmit side of NCP connection protocol.

@RECEIVE FROM HOST # (@R F H #)
is the network address of the host or port. Establishes host # parameter for manual initialization of the connection.

@RECEIVE FROM SOCKET # (@R F S #)
is the socket number in octal. Establishes socket # parameter for manual initialization of the connection.

@SEND TO HOST # (@S T H #)
is the network address of the host or port. Establishes host # parameter for manual initialization of the connection.

@SEND TO SOCKET # (@S T S #)
is the socket number in octal. Establishes socket # parameter for manual initialization of the connection.

As mentioned earlier, one use of the low-level protocol is to open a connection to a socket other than the standard Telnet logger socket. The following example demonstrates how to open a connection to socket 13 on host 1 of IMP 6.

```
@HOST 1/6
@RECEIVE FROM SOCKET 13
@INITIAL CONNECTION PROTOCOL
```

A connection would now exist to the selected socket on the host/IMP.

Another use for the low-level protocol is to open a connection to another TAC port. The socket number of the destination TAC port must be known. The socket number is a 32-bit word. The high order 16-bits contain twice the (octal) port number. The low order 16-bits contain a 2 (or 3) for the send (or receive) socket. The following commands demonstrate how to open a connection to port 3 of the TAC on host 2 of IMP 40:

```
@SEND TO HOST 2/40
@RECEIVE FROM HOST 2/40
@SEND TO SOCK  ET 600002
@RECEIVE FROM SOCKET 600003
```

At this point, the TAC must initiate the connection. Type:

```
@PROTOCOL TO TRANSMIT  (@P T T)
@PROTOCOL TO RECEIVE  (@P T R)
```

Or, to initiate both protocols at the same time, type:

@PROTOCOL BOTH (@P B)

The connection to port 3 of TAC host 2 on IMP 40 is complete.

5. Special Operational Issues

5.1 Dealing with the Network Operations Center (NOC)

The Network Operations Center is located at, and operated by, Bolt Beranek and Newman Inc. in Cambridge, Massachusetts. It is staffed 24 hours a day, seven days a week. Should problems occur that the user is unable to diagnose and fix locally, the NOC can assist by: fixing TAC problems and setting ports; setting up and changing site parameter files; helping to diagnose problems on the TAC, remote host, phone lines or terminals; answering questions, explaining situations and referring problems through the proper channels.

It is helpful to both the user and the NOC if problems with the TAC are reported immediately. When contacting the NOC, the user should try to be as specific as possible. Diagnosing and fixing a problem can be most readily accomplished if the user can tell the NOC what TAC system they are using, and the port number on the TAC. Also, informing the NOC of the time of day the problem occurs can help relate the problem to other network events.

To contact the NOC immediately, telephone (617)661-0100. If the problems or questions are not critical, network mail can be sent to the NOC. The address is CONTROL@BBN-NOC.

5.2 Site Tailoring and Default Values

Each TAC site has a site-specific file that contains parameter information unique to each site. This site file is included whenever the TAC is reloaded by the NOC. The file contains the name of the site, the port configuration, and other information needed by the TAC program.

The site-specific file allows individual ports on a TAC to be tailored for particular devices. This is very helpful when equipment such as lineprinters, special devices (micro-computers, smart terminals), etc. are connected to the TAC.

A liaison at each TAC site is responsible for coordinating and requesting changes to the site file. If users wish to have specific options enabled for their TAC ports, they should contact the site liaison. The NOC can provide the name of the liaison for any site upon request.

The specific parameter options that can be tailored for a TAC port are:

- input and/or output rates
- non-hunting (default is hunting)
- parity (default is none)
- EIA or current loop capability (default is EIA)
- permanent rates (default is changeable rate setting)
- TCP or NCP protocol (default is TCP)
- insert <Linefeed> or <Null> after <Carriage-return> (default is insert <Linefeed>)
- padding (default is no padding)
- flow control enabled (default is flow control disabled)
- echo mode (default is local echo)
- wild mode (default is not wild)
- intercept character (default is "@")
- permanent intercept character (default is changeable intercept character assignment)
- transmission frequency (default is every character)
- low-level protocol authorization (default is not authorized)
- old Telnet authorization (default is not authorized)

In addition to the standard herald message, the TAC can send

a custom herald message to all of its users. When a user closes a connection or hunts to speed on a port, a short message can be sent to the user's terminal along with the normal TAC herald message. This is useful to warn users of a particular TAC of extended outages or dial-up access changes. If this feature is desired, contact the NOC for details.

5.3 Differences Between the TIP and the TAC

Although the TAC operates in a similar manner to the older TIP systems, there are a number of important differences. These are:

- TCP/IP protocols are supported on the TAC.
- The TAC uses a different hunting scheme. It can hunt to higher rates. It can hunt on direct connect or dial-in terminals.
- The TAC does not implement the Telnet Remote Controlled Transmission and Echoing (RCTE) option.
- The TAC does not support the IBM 2741 terminal or the magnetic tape option.
- The TAC does not implement the divert output option.
- Low level protocol commands are only supported for NCP.
- Old format addressing is not supported.
- Some error messages have been changed and made more

descriptive.

- The TAC is a separate host from the IMP. There can be more than one TAC on an IMP.
- The TAC commands are exectued by a <Carriage-return>.
- The reset command does not set a port back to hunting mode.

6. Appendix A - Command Summary

Following is a list of commands available on the TAC, followed by a brief description of use for each command. The commands are presented in capital letters. Lower case words and "#" are parameters. "*" signifies the command is allowed in NCP mode only. (NOTE: only the first letter of each word in a command is recognized by the TAC. It is possible, therefore, to abbreviate the command to be just the first letter of each word. The abbreviated format is given in parentheses.)

@BINARY INPUT END (@B I E)
Turn off 8-bit binary input mode.

@BINARY INPUT START (@B I S)
Turn on 8-bit binary input mode.

@BINARY OUTPUT END (@B O E)
Turn off 8-bit binary output mode.

@BINARY OUTPUT START (@B I S)
Turn on 8-bit binary output mode.

@CLEAR DEVICE WILD (@C D W)
Take device out of wild mode.

@CLEAR INSERT LINEFEED (@C I L)
Stop inserting <Linefeed> after <Carriage-return>.
Inserts a <Null> after <Carriage-return>.

@CLOSE (@C)
Close the connection.

@DEVICE CODE ASCII (@D C A)
No padding performed on output.

@DEVICE CODE EXTRA-PADDING (@D C E)
Insert padding on output for terminals with slow carriage-return.

@DEVICE CODE OTHER-PADDING (@D C O)
Insert padding in output for lineprinter.

@DEVICE CODE 37 (@D C 37)
The TAC sets the parity of the port to even parity.

@DEVICE RATE # (@D R #)
Set input and output rates and character size to #.

@ECHO HALFDUPLEX (@E H)
TAC echoes nothing; terminal generates echo. (The TAC inserts the <Linefeed> character.)

@ECHO LOCAL (@E L)
TAC generates echo.

@ECHO REMOTE (@E R)
Remote host echoes data; TAC echoes TAC commands.

@FLOW-CONTROL INPUT END (@F I E)
Flow control from terminal to TAC is disabled.

@FLOW-CONTROL INPUT START (@F I S)
Flow control from terminal to TAC is enabled.

@FLOW-CONTROL OUTPUT END (@F O E)
Flow control from TAC to terminal is disabled.

@FLOW-CONTROL OUTPUT START (@F O S)
Flow control from TAC to terminal is enabled.

@FLUSH (@F)
Delete all unsent characters in input buffer.

@# GIVE BACK (@# G B)
Give up control of port #. (# is the octal port number.)

* @HOST # (@H #)
  Simultaneously sets "@S T H" and "@R F H". # is the host/imp address.

* @INITIAL CONNECTION PROTOCOL (@I C P)
  Starts the NCP initial connection protocol.

  @INSERT LINEFEED (@I L)
  Insert <Linefeed> after <Carriage-return>.

  @INTERCEPT # (@I #)
  Set intercept character to #. # is a decimal number representing an ASCII character.

  @INTERCEPT ESCAPE (@I E)
  Reset the intercept character back to the default "@".

  @INTERCEPT NONE (@I N)
  Enables 7-bit binary mode.

  @NEW TELNET (@N T)
  Enables new Telnet mode.

  @OLD TELNET (@O T)
  Enables old Telnet mode. Can only be used with permission from the NOC.

  @OPEN net#:host#/imp#;port# (@O net#:host#/imp#;port#)
  Open a connection to a specific host on the specified network. This may be used for TCP or NCP connections. ("net#:" is not specified for NCP.) Authorization required to set the ";port#" parameter for the receiving port.

  @OPEN a.b.c.d;port# (@O a.b.c.d;port#)
  Open a connection to a specified internet address. This is used for TCP connections only. Authorization required to set the ";port#" parameter for the receiving port.

* @PROTOCOL BOTH (@P B)
  Simultaneously sets NCP "@P T T" and "@P T R" commands. Authorization required for the

receiving port.

@PROTOCOL NCP (@P N)
Sets the protocol to NCP.

@PROTOCOL TCP (@P T)
Sets the protocol to TCP.

* @PROTOCOL TO RECEIVE (@P T R)
Initiates receive side of NCP connection protocol. Authorization required for the receiving port.

* @PROTOCOL TO TRANSMIT (@P T T)
Initiates transmit side of NCP connection protocol. Authorization required for the receiving port.

* @RECEIVE FROM HOST # (@R F H)
Establishes host # parameter for manual initialization of connection protocol. # is the network address of the host.

* @RECEIVE FROM SOCKET # (@R F S #)
Establishes socket # parameter for manual initialization of connection protocol. # is the socket number in octal.

* @RECEIVE FROM WILD (@R F W)
Same as "@R F S <socket #>".

@RESET (@R)
Resets the TAC port. Resets existing connection.

@SEND ABORT OUTPUT (@S A O)
Send the Telnet "AO" command.

@SEND ARE YOU THERE (@S A Y T)
Send the Telnet "AYT" command.

@SEND BREAK (@S B)
Send the Telnet "BRK" command.

@SEND ERASE CHARACTER (@S E C)
Send the Telnet "EC" command.

@SEND ERASE LINE (@S E L)
Send the Telnet "EL" command.

@SEND INTERRUPT PROCESS (@S I P)
Send the Telnet "IP" command followed by the SYNC character sequence. (Not yet implemented in TCP.)

@SEND SYNC (@S S)
Send the Telnet "SYNC" character and an "Interrupt Process" message.

* @SEND TO HOST # (@S T H #)
Establishes the host # parameter for manual initialization of connection protocol. # is the network address of the host.

* @SEND TO SOCKET # (@S T S #)
Establishes the socket # parameter for the manual initialization of connection protocol. # is the socket number in octal.

@SET DEVICE WILD (@S D W)
Port will accept a connection from any host. Authorization required for the receiving port.

@TRANSMIT EVERY # (@T E #)
The TAC will attempt to send every # characters to the remote host. # is a decimal number.

@TRANSMIT NOW (@T N)
The TAC immediately sends all characters stored in the input buffers to the remote host.

@TRANSMIT ON LINEFEED (@T O L)
The TAC transmits to the remote host whenever a <Linefeed> is received from the terminal.

@TRANSMIT ON MESSAGE-END (@T O M)
The TAC transmits to the remote host whenever a <Control-S> is received from the terminal.

@TRANSMIT EVERY 0 (@T E 0)
Resets TRANSMIT ON MESSAGE-END and TRANSMIT ON LINEFEED modes.

7. Appendix B - TAC Messages to the Terminal User

Following is a list of messages the TAC may, at times, give the terminal user.

<sitename> TAC <version#>:<port#>
The TAC has acknowledged the user attempt to connect to the TAC. <version#> is the software version number running in the TAC. <port#> is the octal port number the user is connected to. The user can now use the TAC to connect to a remote host.

Bad
The TAC does not recognize the command.

Can't
The TAC could not execute the command.

Closed
The TAC's connection to the remote host is closed.

Destination host dead
The remote host is not communicating with the network.

Destination unreachable
There is no path from the TAC to the remote host through the communication networks.

Host closing connection
The remote host has closed its connection to the TAC. The TAC will close its connection to the host. The port will then be idle.

Host down until <day> at <hour>:<minutes> <timezone>.
The remote host is not communicating with the network. The day and time when the host was most recently scheduled to come up is indicated.

Host reset connection

The remote host has reset the connection to the TAC.

NCP Trying...
The TAC is attempting to open a NCP connection.

No
Parameters cannot be set for the specified port.

Not authorized
Low-level protocol authorization is needed. Contact the NOC.

Num
The TAC expects a number. The command is aborted.

Open
The TAC's connection to the remote host is open.

Open error
An error occurred while the OPEN attempt was in progress. This probably indicates a host error and should not happen often. If it is recurrent, contact the NOC (see Section 5.1).

Refused
The remote host rejected the attempt to establish a connection. This may occur if the remote host does not support TCP and/or Telnet. The TAC port will now be idle.

Retransmitting
This indicates that TCP has to retransmit many times to open a connection to a remote host, or to get TAC data accepted by that host. The message will occur after TCP has retransmitted five times. It will appear about once a minute until the data is accepted, or the user resets the connection.

Set Input Rate, Then Type ^Q
The TAC has hunted to an acceptable output rate, but too high an input rate. The user must set the input rate to 2400 baud or less, then type <Control-Q>. (This applies only to H-316 TACs.)

TAC's IMP going down in <mins> mins for <hrs> hrs <mins> mins.
The TAC's IMP is going down in the time indicated. Although the TAC will still respond, it will be isolated from the network.

TAC's IMP going down NOW
The TAC's IMP is going down immediately.

TAC's IMP down
The TAC's IMP is down. The TAC is isolated from the network.

TAC going down in <mins> mins for <hrs> hrs <mins> mins.
The TAC is going down in the time indicated.

TAC going down NOW
The TAC is going down immediately.

TCP Trying...
The TAC is attempting to open a TCP connection to a remote host.

8. Appendix C - Connection of Terminals to the TAC

The TAC permits a wide variety of terminal types to access it. The basic requirement for the TAC is that the terminals and modems using it conform to EIA RS-232 specification. It will also support terminals that use 20 milliampere current loop to communicate.

The TAC allows for the use of modems connected via dedicated or dial-up lines. A variety of manufacturers' modems will work on the TAC. The requirements for these modems are that they be similar to Bell 103, Bell 202S and 202T, and Vadic 3400 series modems.

When selecting modem equipment for use on the TAC, it is advised that the user contact the NOC for assistance. There are many modem manufacturers, and selection of a proper modem will make using the TAC more convenient.

In any case, if there is a question about proper equipment for TAC use, contact the NOC. They will be able to help the user in determining what can be best applied to the TAC.

The tables on the following pages list the pin and signal allocations for TACs with connectors for modems, EIA terminals and current loop terminals.

PIN	SIGNAL DESIGNATION
1	Protective Ground
2	Transmitted Data (TD) (from TAC)
3	Received Data (RD) (to TAC)
4	Request to Send (RTS) (from TAC)
6	Data Set Ready (DSR) (to TAC)
7	Signal Ground
8	Data Carrier Detect (DCD) (to TAC)
9	+ VDC
10	- VDC
15	Transmitted Clock (TC) (to TAC)
17	Received Clock (RC) (to TAC)
20	Data Terminal Ready (DTR) (from TAC)

Table 1. TAC Signal Allocation for EIA RS-232 Modem Connector

The EIA RS-232 modem connector can be used with a modem or a terminal. Table 1 lists the pin assignments for a modem connected via a cable with no crossings or jumpers - all signals should run straight through.

If a terminal is to be connected to the port, the cable must be modified to be a null modem cable. To do this: cross the Data Set Ready (DSR) pin of the fantail connector to the Data Terminal Ready (DTR) pin of the terminal connector; cross the DTR pin of the fantail connector to the DSR pin of the terminal connector; cross the Request to Send (RTS) pin of the fantail connector to the Data Carrier Detect (DCD) pin of the terminal connector; cross the DCD pin of the fantail connector to the RTS pin of the terminal connector; and, jumpering pin 4 to pin 5 in the terminal connector. A null modem adapter is available to accomplish the above.

PIN	SIGNAL ALLOCATION
1	Protective Ground
2	Transmitted Data (TD) (by terminal)
3	Received Data (RD) (by terminal)
4	Request to Send (RTS) (from terminal)
5	Clear to Send (CTS) (from TAC)
6	Data Set Ready (DSR) (from TAC)
7	Signal Ground
8	Data Carrier Detect (DCD) (from TAC)
9	+ VDC
10	- VDC

Table 2. TAC Signal Allocation for EIA RS-232 Terminal Connector

The EIA RS-232-C terminal connector is designed for terminals with an EIA RS-232 interface. No crossed signals or jumpers are needed.

PIN	SIGNAL DESIGNATION
1	Spare
2	Transmitter Source (TS)
3	Ground
4	Receiver Return (RR)
5	Spare
6	Spare
7	Transmitter Return (TR)
8	Receiver Source (RS)
9	Spare

Table 3. TAC Signal Allocation for Current Loop Cable Connector

Current loop connectors are for asynchronous 20ma current loop terminals. Current source is provided by the system for all signals. Cable wiring is dependent on the connector pin assignments at the terminal to be used.

9. Appendix D - Device Rate Manipulation

In section 4.2 - "Device Rate" - examples were given for various device rates using 8-bit characters. This appendix will give the user the breakdown of the device rate parameter to allow for character size different from 8-bit. It also gives the user the formula to set a variety of input and output rates.

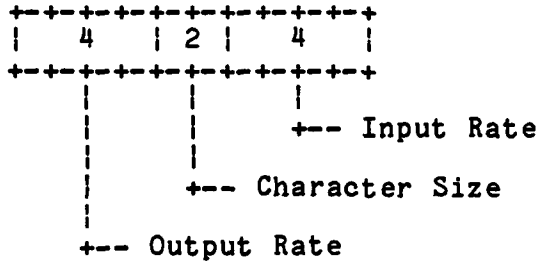
It should be noted that certain hardware limitations exist on both the C/30 TAC and the H-316 TAC. The C/30 TAC requires the terminal connected to a port be set to the same input/output rate. Legal rates are: 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2400, 4800 and 9600 baud. A terminal may hunt to any of these rates (with the exception of 9600/9600 baud. Another character, <Carriage-return> for example, can be used to hunt instead of <Control-Q>). Also, the C/30 TAC will only recognize 8-bit character size.

For the H-316 TAC, the hardware limitation prohibits input rates to be greater than 2400 baud. The H-316 TAC, therefore, has the capability to have different input and output rates. See Sections 2.1.2 - "Split Rate Hunting" - and 2.1.3 - "Device Rate Command" - for further discussion.

When setting a device rate for a port the user types:

@DEVICE RATE # (or, @D R #)

represents, in decimal, a 10-bit field. The field is divided as follows:



To determine the device input/output rate and the character size, use the following table.

Character Size		Input/Output Rates	
0 (00)	5-bits	0	hunting
1 (01)	6-bits	1	75 baud
2 (10)	7-bits	2	110
3 (11)	8-bits	3	134.5
		4	150
		5	300
		6	600
		7	1200
		10	1800
		11	2400
		12	4800 (output only for H-316)
		13	9600 (output only for H-316)
		17	synchronous

10. Appendix E - Internet Addressing: Non-ARPANET-type Networks

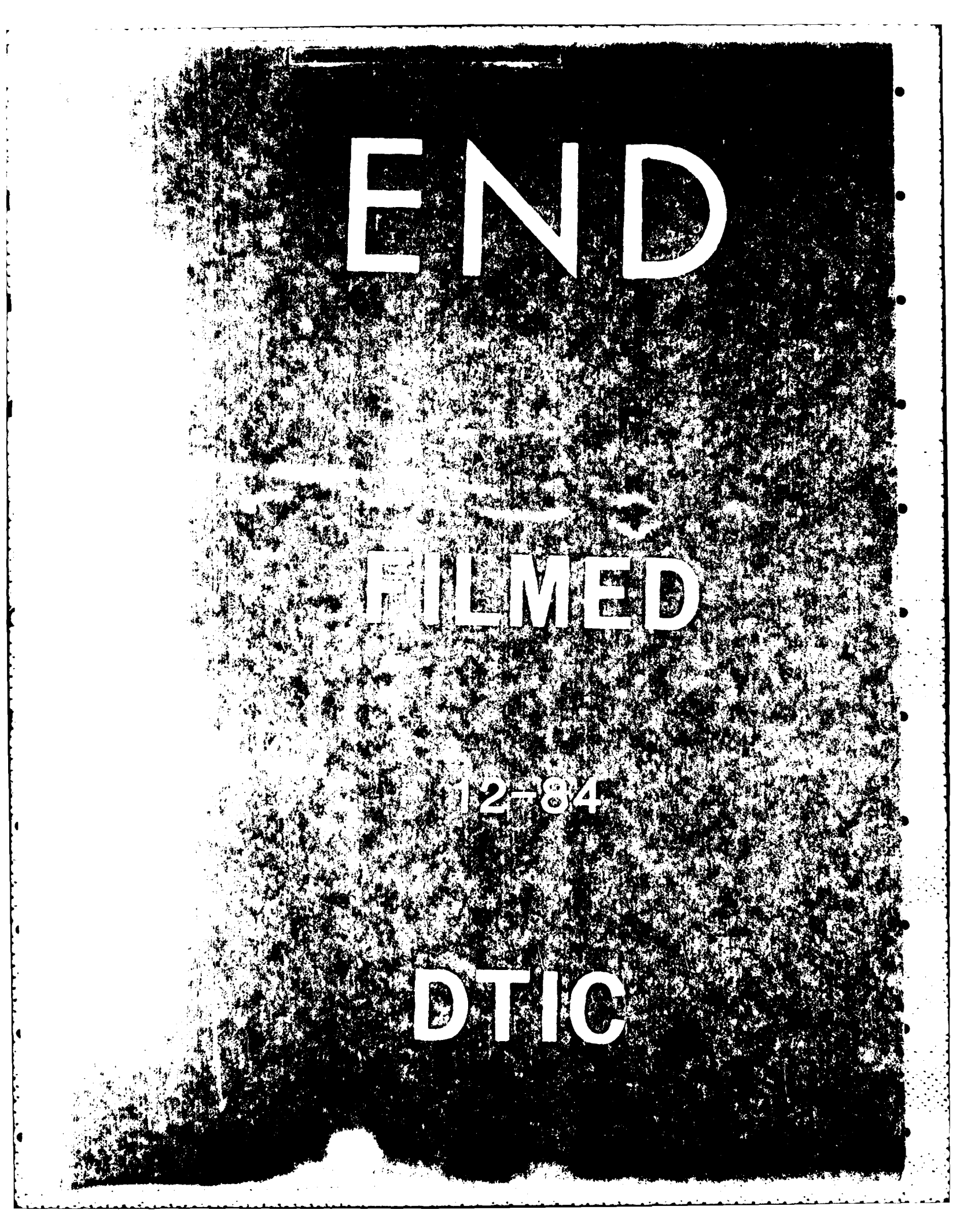
The internet addressing scheme makes provision for three classes of networks: classes A, B and C. The ARPANET is a class A network. The TAC expects to be connected to a class A network. The TAC is, however, capable of communicating with host computers on the other classes of networks. The main difference in addressing the different classes of networks is in the interpretation of the four bytes of the internet address.

For class A networks: the first byte is interpreted as the network number. The following three bytes are interpreted as a local address. Class A network numbers may range from (decimal) 0 through 127. For example, to connect to host 3 on IMP 49 on net 10, the address would be 10.3.0.49.

For class B networks: the first TWO bytes are the network number. The remaining two bytes are the local address. The first byte of a class B network must be between 128 and 191, and the second byte may take any value between 0 and 255. An example of a class B internet address is 128.10.0.2.

For class C networks: the first THREE bytes are the network number. There is only one byte of local address. The first byte of a class C network must have a value between 192 and 223, and the second and third bytes may take on any value between 0 and 255. An example of a class C internet address is 192.1.2.67.

To access a class B or class C network, or a class A network which uses the third byte of the internet address in its local address (unlike the ARPAnet, which expects this byte to be zero), it is necessary to use the a.b.c.d format for the Open command on the TAC.


END
FILMED
12-84
DTIC